UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/820,295	04/08/2004	Brian Connell	22493-27U(16666ROUS01U)	7444

31292 7590 03/16/2010

CHRISTOPHER & WEISBERG, P.A.
200 EAST LAS OLAS BOULEVARD
SUITE 2040
FORT LAUDERDALE, FL 33301

EXAMINER	
MITCHELL, JASON D	
ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
03/16/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/820,295
Filing Date: April 08, 2004
Appellant(s): CONNELL ET AL.

Alan M. Weisberg (reg. no. 43,982)
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 12/17/09 appealing from the Office action mailed 7/21/09.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

The amendment after final rejection filed on 12/14/09 has not been entered.

The examiner acknowledges that the amendment to claim 21 would, if entered correct the 35 USC §112 2ND issue. However, the amendment was filed on the same day as the appeal brief and does not fall within the category of amendments allowed entry when filed on or after the data of filing an appeal brief (see 37 CFR 41.33(b)(1-2)).

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

GROUNDS OF REJECTION NOT ON REVIEW

The following grounds of rejection have not been withdrawn by the examiner, but they are not under review on appeal because they have not been presented for review in the appellant's brief.

35 USC § 112 2nd rejection of claim 21. The examiner agrees that the amendment filed on 12/14/09 would overcome this rejection if entered. The examiner will enter the amendment if all other patentability issues are decided in the appellants' favor. Accordingly, the board need not address this issue.

(7) Claims Appendix

A substantially correct copy of appealed claims 1-29 appears on page A of the Appendix to the appellant's brief. The minor errors are as follows: the text of claim 21 as presented in the appendix contains changes to the dependency made in the amendment filed on 12/14/09 and not entered by the examiner.

(8) Evidence Relied Upon

20020091809	Menzies et al.	7-2002
20030046370	Courtney	3-2003

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1-4, 8-10, 12-16, 18-22, 24, 26-27 and 29 are rejected under 35 U.S.C. 102(b) as being anticipated by US 2002/0091809 to Menzies et al. (Menzies).

Regarding Claims 1, 13 and 19: Menzies discloses a method comprising:

modeling network element commands (par. [0089] “a SNMP GetNext function”), events (par. [0074] “events are modeled as TRAP-TYPE and NOTIFICATION-TYPE macros”) and run-time system data (par. [0066] “MIB contains two types of structures for describing dynamic device data”) into a data model using a first modeling language, the data model comprising first data (par. [0064] “each network device provides information according to the SNMP standard … i.e. via MIBs”; also see par. [0004] “describe the data … using an appropriate MIB module to model the data”);

translating the first data represented in the first modeling language to second data represented in a second modeling language (par. [0064] “maps MIB information to CIM classes”);

storing the second data in the second modeling language in a global data model repository ([0039] “The Common Information Model (CIM) defines the model used to represent the real-world objects being managed”); and

automatically generating code to support an external management interface based on the stored second data in the global repository, the external management interface communicating with the stored second data (par. [0060] “transparently translate the user’s high-level query into a series of simple retrievals and then perform the query internally on behalf of the user”).

In the case of claim 13, Menzies further discloses

a memory comprising a global repository (Fig. 3, CIM Repository 74; Fig. 1 RAM 25);

a processor electrically coupled to the memory (Fig. 1, Processing Unit 21);

a first interface to a plurality of network elements in communication with the global repository (Fig. 3, COM, SNMP, RPC, HMMP and WIN32 communication with Objects 76_{1-n}); and

a second interface to an external interface in communication with the global repository (Fig. 3, DCOM or HMMP communication with Management Applications 58_{1-n}).

Regarding Claims 2, 14 and 20: The rejections of claims 1, 13 and 19 are incorporated; further Menzies discloses automatically generating system documentation based on the stored second data (see e.g. the table following par. [0048] "[description ("This class models a disk")]").

Regarding Claims 3, 15 and 21: The rejections of claims 2, 14 and 20 are incorporated; further wherein the generated system documentation corresponds to code generated to support an external management interface (par. [0048] "[description ("This class models a disk")]").

Regarding Claims 4, 16 and 22: The rejections of claims 1, 13 and 19 are incorporated; further Menzies discloses the first modeling language is structured management information (SMI) (par. [0070] "the SMI compiler 88 for processing the MIB tree structure").

Regarding Claims 8: The rejection of claim 1 is incorporated; further Menzies discloses automatically generating code for the external interface includes automatically generating code to implement a Simple Network Management Protocol interface (par. [0063] "Common Information SNMP Mapping").

Regarding Claims 9 and 26: The rejections of claims 1 and 19 are incorporated; further Menzies discloses automatically generating code for the external interface includes automatically generating code to implement a configuration database (par. [0038] "the CIM database 74").

Regarding Claims 10 and 27: The rejections of claims 1 and 19 are incorporated; further Menzies discloses automatically generating code for the external interface includes automatically generating code to implement Simple Network Management Protocol SNMP subagents (par. [0004] "agents provide such management information via a standard known as the Simple Network Management Protocol (SNMP)"; par. [0063] "Common Information SNMP Mapping").

Regarding Claims 12, 18, 24 and 29: The rejections of claims 1, 13 and 19 are incorporated; further Menzies discloses modeling the run-time system data from a plurality of sources using at least one of the first modeling language and the second

modeling language (par. [0035] "access a CIM repository 74 in order to determine which object provider or providers to contact").

Claims 5-7, 11, 17, 23, 25 and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 2002/0091809 to Menzies et al. (Menzies) in view of US 2003/0046370 to Courtney (Courtney).

Regarding Claims 5, 17 and 23: The rejections of claims 1, 13 and 19 are incorporated; further Menzies does not disclose the second modeling language is extensible markup language (XML).

Courtney teaches a method of abstracting network elements wherein the second modeling language is XML (par. [0036] "convert the active command format for the network device 165 into an XML … format").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to model Menzies' second data ([0039] "The Common Information Model (CIM) defines the model used to represent the real-world objects being managed") in XML as taught by Courtney (par. [0036] "convert the active command format for the network device 165 into an XML … format"). Those of ordinary skill in the art would have been motivated to do so as a known alternate method for storing the data which would have provided only the expected functionality (Menzies par. [0060] "transparently

translate the user's high-level query into a series of simple retrievals and then perform the query internally on behalf of the user"; par. [0038] "the system administrator 175 is presented with network device configurations in a standard, consistent format"; Courtney par. [0037] "One advantage of the DOM format is that is provides a standard format for most ne5twork device configurations").

Regarding Claims 6, 7 and 25: The rejections of claims 1 and 19 are incorporated; further Menzies does not explicitly disclose automatically generating code to implement a command line interface (CLI) or an XML interface.

Courtney teaches automatically generating code to implement a command line interface (CLI) (par. [0040] "the XML-CLI converter 200 allows the system administrator 175 to interface with CLI-based network devices") and an XML interface (par. [0039] "the standard XML-to-native XML converter").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Menzies' methods (par. [0064] "maps … information to CIM classes") to generate code to implement a CLI or XML interface as taught by Courtney (par. [0040] "the XML-CLI converter 200 allows the system administrator 175 to interface with CLI-based network devices"; par. [0039] "the standard XML-to-native XML converter"). Those of ordinary skill in the art would have been motivated to do so in order to expand the applicability of Menzies' methods (see e.g. Menzies par. [0036] "gather the

necessary data from the devices ... using vendor or protocol-specific mechanism ... or a proprietary mechanism“).

Regarding Claims 11 and 28: The rejections of claims 1 and 19 are incorporated; further Menzies discloses generating a class repository (par. [0059] “CIM classes … stored in the CIM repository”) but does not explicitly disclose automatically generating code to assist in implementation of an Application Program Interface.

Courtney teaches automatically generating code to assist in implementation of an Application Program Interface (par. [0043] “The DOM applications can also include an (API)”).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to implement Menzies’ class repository (par. [0059] “CIM classes … stored in the CIM repository”) as an API as taught by Courtney (par. [0043] “The DOM applications can also include an (API)”). Those of ordinary skill in the art would have been motivated to do so as a known alternative method of providing the disclosed functionality with would have produced only the expected results (Menzies par. [0060] "transparently translate the user's high-level query into a series of simple retrievals"; Courtney [0039] “Another application that utilizes the DOM … use the same … command format for most network devices even though each device”).

(10) Response to Argument

A. The Rejection of Claims 1-4, 8-10, 12-16, 18-22, 24, 26-27 and 29 Under 35 USC §102(b)

In the par. bridging pp. 4-5, the appellants state:

> The data model in the present invention and as recited in Claims 1, 13 and 19 is more than just a configuration file, as it includes commands, events (such as alarms, statistics, counters, etc.) and run-time system data. For example, the network element gathers network element data at run-time from multiple components or software processes that comprise the network element. This operational run-time data provides network operators the ability to keep, for instance, a router's statistics and accounting data, as well as information needed to manage the router. The network element, e.g. the router, is managed through "commands, events and run-time system data" to provide, among others, maintenance and billing information. This arrangement is not taught or suggested by Menzies.

The examiner respectfully disagrees. First it is noted that only the data model in the first modeling language explicitly “includes commands, events ... and run-time system data". Further, Menzies discloses a data model in a first modeling language (par. [0064] “information according to the SNMP standard … i.e. via MIBs”; also see par. [0004] “describe the data … using an appropriate MIB module to model the data”) and that this model includes commands (par. [0089] “a SNMP GetNext function”), events (par. [0074] “events are modeled”) and run-time data (par. [0066] “MIB contains … dynamic device data”). Note that no specific details regarding the commands, events, run-time data or how the modeling languages represent them are recited in the claims. Accordingly the cited modeling of commands, events and run-time data meets the broadly claimed limitations.

In the par. bridging pp. 5-6 the appellants state:

> Initially, Menzies fails to disclose "translating the first data represented in the first modeling language to second data represented in a second modeling language". The Examiner cites paragraph [0064] in Menzies for allegedly disclosing this feature. However, Menzies merely maps management information from one schema (MIB) to another (CIM). FIG. 5 of Menzies shows how MIB information is mapped to CIM classes. In the CIM installation, each MIB is extracted from an SNMB MIB repository (SMIR) and translated by an SMI (Structure of Management Information) compiler into a CIM object class, which is provided to the CIM object manager CIMOM and stored in the CIM repository. To accomplish this, the SMI compiler references a number of mapping tables to facilitate the translation. The CIMOM combines the dynamic information with the static MIB information now in the form of CIM classes to produce a desired instance output. Menzies does not "model network element commands, events and run-time system data into a data model using a first modeling language" or "translate the first data represented in the first modeling language to second data represented in a second modeling language". Claims 1, 13 and 19 are believed patentable over Menzies for at least this reason.

The examiner respectfully disagrees. First it is not clear what distinction the appellants are attempting to make here because they appear to acknowledge Menzies discloses the claimed translation (e.g. "each MIB is ... translated ... into a CIM object class"). To the extent that the appellants are arguing that Menzies' MIBs and CIM objects are not defined in a "modeling language", the examiner notes that the MIBs are defined by macros (see e.g. the table shown in par. [0067]) and that CIM is disclosed as "classes, instances, and qualifiers … described in a user-intuitive textual format" (see par. [0005]). Those of ordinary skill in the art would understand that such models meet the broadly claimed "modeling language" aspects of the claims.

In the first full par. on pg. 6 the appellants state:

> The Examiner also relies on Menzies's ability to "transparently translate the user's high-level query into a series of simple retrievals and then perform the query internally on behalf of the user" (paragraph [0060]) as somehow equating with Appellants' "automatically generating code to support an external management interface based on the stored second data in the global repository, the external management interface communicating with the stored second data". However, Menzies is not "automatically generating code to support an external management interface based on the stored second data in the global repository". Instead, Menzies is merely taking a user's high-level query, such as an SQL query, and translating the query into a series of simpler queries, "giving the impression that the provider directly supported the query". (Menzies, paragraph [0060]). Menzies does not automatically generate code based on stored second data in a global repository. Claims 1, 13 and 19 are believed patentable over Menzies for at least this additional reason.

The examiner respectfully disagrees. The claims only broadly recite "automatically generating code … based on the stored second data", and do not provide any details regarding what is generated or how it is generated. Menzies' programmatic generation of "simpler queries" (see e.g. par. [0060] "the CIMOM 70 ... complements the capabilities of the provider … translates the user's high-level query into a series of simple retrievals") meets the broadly claimed limitation. More specifically, those of ordinary skill in the art would understand Menzies' translation (par. [0060]) to describe the 'generation' of "a series of simple retrievals" which were not provided by the user. Further, those of ordinary skill in the art would understand that these "simple retrievals" describe commands (or more broadly, code) to be submitted to the providers which will then process the commands and return the desired information to be combined by the CIMOM (using more generated code) and then returned to the user. Accordingly, Menzies disclosure meets, at least, a reasonably broad interpretation of the claims.

B. The Rejection of Claims 5-7, 11, 17, 23, 25 and 28 under 35 USC §103(a)

In the first par. on pg. 8, the appellants state:

> Courtney fails to disclose the features of Appellants' claims that are not taught or suggested by Menzies. Courtney merely deals with configuring network equipment by simply converting a router's commands from a native representation into a standard format. Courtney does not teach an interface that interacts and communicates with a stored data model, because Courtney does not store a data model at all. Courtney says nothing about a data model that models a network element's commands, events, and run-time system data. Courtney merely creates commands in a standard format to interface with the routers themselves, not with a data model representation of the router. In fact, nothing in Courtney teaches the concept of a complete data model representation of a network element, where the data model itself is used to automatically generate code for an interface that communicates and interacts with the stored data model.

The examiner respectfully disagrees. First it appears that the appellants' are only arguing that Courtney does not teach the claim limitations anticipated by Menzies (see the discussion above). But regardless, Courtney teaches an XML based modeling language (par. [0036] "convert the active command format for the network device 165 into an XML and/or DOM format"; also see par. [0016] "model a network device's configuration by … converting it into … an XML document or a DOM"). Courtney's DOM is merely an alternate method of modeling a network device's configuration and those of ordinary skill in the art would have understood that it could be substituted for Menzies' second model of a network device ([0039] "The Common Information Model (CIM)") to some benefit (see e.g. Courtney par. [0037] "One advantage of the DOM format is that is provides a standard format for most network device configurations").

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Jason Mitchell/

Primary Examiner, Art Unit 2193

Conferees:

/Lewis A. Bullock, Jr./

Supervisory Patent Examiner, Art Unit 2193

/Hyung S. Sough/

Supervisory Patent Examiner, Art Unit 2194

03/12/10